PROJECT REPORT ON

DIGITAL VALUATION SYSTEM

SUBMITTED BY

Mr. PRAMOD SREERAM YADAV (1059269)
Mr. ROHIT VIJAYKUMAR YADAV (1059267)

UNDER THE GUIDANCE OF

Prof. Mrs. BHARATI .BHOLE

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR QUALIFYING B.Sc. (I.T),
SEM ESTER-VI EXAMINATION

HINDI VIDYA PRACHAR SAMITI'S

RAMNIRANJAN JHUNJHUNWALA COLLEGE,

OPPOSITE GHATKOPAR RAILWAY STATION,
GHATKOPAR (W), MUMBAI 400 086

YEAR 2015-2016

University Of Mumbai

Project Certificate

This is to certify that Project entitled _____ undertaken at the _____ by Mr. / Ms _____ Seat no _____ in partial fulfilment of B.Sc. IT degree (Semester VI) Examination had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

It is further certified that he / she has completed all required phases of the project.

Signature of Internal Guide

Sign of coordinator

College Seal

Sign of Examiner

PREFACE

For every evaluator ,correcting paper is one of the most important thing. Each and every evaluator have to travel to the examination center to evaluate the papers for this university have to spend money on transportation of paper as well as evaluators.
Due to this a time and cost is being wasted.Time is precious so Digital Valuation System (DVS) have been proposed.

DVS is a system which is built to ease the work and saves the time and cost which are spent on transportation. Lots of time were spent on travelling so their was a delay in declaration of result due to which student have to wait for a long time.
This system helps evaluator as well as student to get fast and accurate result. Manual work was tedious to do.Making digital helps to solve all the problems which were faced by evaluators.

ACKNOWLEDGEMENT

We would like to add few words for people who have helped in bringing out the creativeness of this project.

To commence with the things we would like to first humbly thanks **Mrs. Dr. USHA MUKUNDAN**, the Principal and **Mrs. BHARTI BHOLE** and **Mrs. ARCHANA BHIDE**, Head of the department of BSC-IT for being appreciative.

We are grateful to our coordinator teacher **Mrs. ZEBA ROSELET** for her encouragement and suggestion on the project. Also thanks to **Mr. SAMBHAJI** who has always more than willing to help us whenever we required any immediate assistance.

Finally, I think my family members and friends for their helpful suggestions and inspiring ideas which were very helpful for this project.

Pramod Yadav

Rohit Yadav

SYNOPSIS

Everything in this era getting digital.Why should valuation should be manual.
Digital Valuation System(DVS) is a software is makes the valuation digital.It gives convenient way to evaluate the papers without wasting time on transportation.Time,Money is saved by using our DVS system.
Due to DVS,Evaluation of paper become fast and easy and hence result declaration can be early as compare to manual process. There is no much computation problem because DVS system compute it automatically.
As the Research is been Implemented in C-SHARP (C#) so , it Provides a great user Friendly Graphical User Interface with a Different Menu's with a Validation features.

In short "DVS is a system which provide a convenient to evaluator "

CONTENTS

SYSTEM REQUIREMENTS SPECIFICATION

1.1. User Requirements:

Application will be accessed through a installing application. The interface would viewed best using 1024 X 720 and 800 X 600 pixels resolution setting. No user would be able to access any part of the application without logging on to the system.

1.2. Hardware Requirements:

- Admin Side:

  Operating System : windows 7 , 8 , 8.1 , 10
  Processor : Pentium 3.0 GHz or higher
  RAM : 1 GB or more
  Hard Drive : 10 GB more

- Evaluator Side:

  Operating System : Windows 7 , 8 , 8.1 , 10
  Processor : Pentium 3.0 GHz or higher
  RAM : 1 GB or more
  Hard Drive : 10 GB more

1.3. Software Requirements:

- Admin Side:

  Microsoft Visual Studio 2015, Framework 4.5.2, Sql server , Adobe Reader

- Evaluator Side:

  Adobe Reader DC ,Framework 4.5.2

1.4. Technology Used:

- **Microsoft SQL Server 2008**:

➢ Microsoft SQL Server 2008 provides the database developer the opportunity to locate programming code in native formats or create server-driven complex system. The .NET Framework, in combination with the SQL server 2008, delivers the functionality of the powerful class library in combination with modern languages.

➢ Common Language Runtime (CLR) is used to code procedures, triggers and functions within the chosen language in the .NET framework.

➢ Using object-oriented constructs such as structured exception handling, namespaces, classes and arrays assist the programmer in effectively handling procedural issues when programming with QL Server 2008.

- **Microsoft Visual Studio 2015**:

➢ Microsoft Visual Studio is an integrated development environment from Microsoft .

➢ Visual Studio supports different languages by means of language services, which allow the code editor and debugger to support nearly any programming language.

➢ It also supports XML/XSLT, HTML/XHTML, javascript and css.

➢ Support for other languages such as M, Python and Ruby among others is available via language services.

1.5 Functional Requirements

Admin:

1. The admin has the full fledge rights over the DVS.
2. Admin upload question papers and answersheet.
3. A admin have previlage to change the role of evaluator, can view the marks.
4. Admin can add or delete the evaluator and provide result.
5. Admin can send reminder to evaluator for checking answersheet.

Evaluator:

1. Evaluator evaluate the answersheet.
2. Evaluator can change the password.
3. Evaluator can view allocation summary.

1.6 Non Functional Requirements

Performance Requirement :

Some Performance requirements identified is listed below:
The Database shall be able to accommodate a minimum of 10000 records of answersheet.
The Software shall support use of multiple users at a time.
There are no other specific performance requirements that will affect Development.

Safety Requirement :

The database may get crashed at any certain time due to virus or operating system failure. Therefore , it is required to take the database backup.

ANALYSIS

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system.

- During analysis, data were collected for the system.
- Training, experience and common sense are required for collection of the information needed to do the analysis.

Once the analysis is completed, there was a firm understanding of what is to be done next. The next step was to decide how the problem might be solved.

Thus, in a system design, we move from a logical to the physical aspect of the life cycle.

So, the Analysis phase fully deals with the collection of the required data from the different sources and analyzing their effect on the intended candidate system.

The Analysis keeps track of the processing occurring within the candidate system and the output generated. After the analysis work comes the design phase where we actually start designing our proposed system.

COST AND BENEFIT ANALYSIS:

- Cost-Benefit Analysis (CBA) estimates and totals up the equivalent money value of the benefits and costs to the community of projects to establish whether they are worthwhile.
- .In order to reach a conclusion as to the desirability of a project all aspect
- of the project, positive and negative, must be expressed in terms of a common unit; i.e. there must be a “bottom line”.
- The most convenient common unit is money. This means that all benefits and costs of a project should be assured in terms of their equivalent money value.
- A program may provide benefits which are not directly expressed in terms of dollars but there is some amount of money the recipients of the benefits would consider as good as projects benefit.

METHODOLOGY INVOLVED

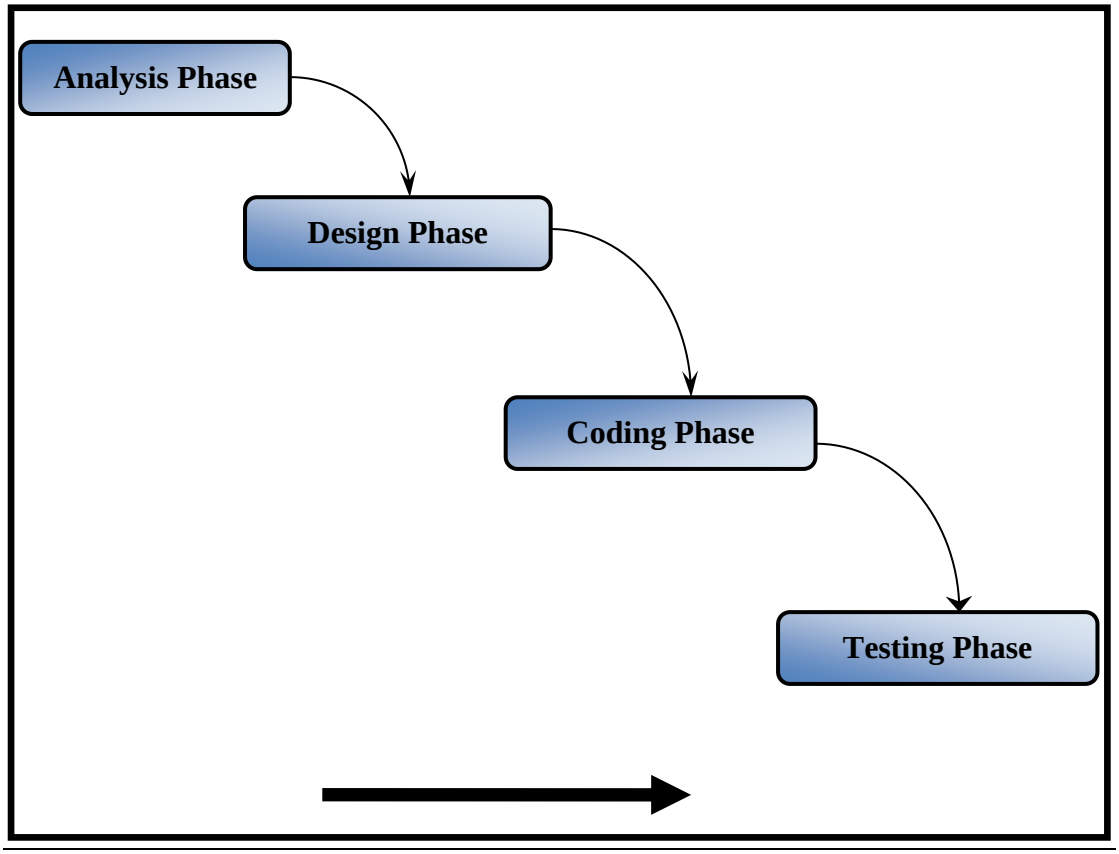
3.1Software Process Model:

Waterfall model:

Software Development Model Used For:Waterfall Model:

Software process model deals with the model which we are going to use for the developmentof the project. There are many software process models available but while choosing it weshould choose it according to the project size that is whether it is industry scale project or big scale project or medium scale project.
Accordingly the model which we choose should be suitable for the project as the software process model changes the cost of the project also changes because the steps in each software process model varies.
This software is build using the waterfall mode. This model suggests work cascading from step to step like a series of waterfalls. It consists of the following steps in the following manner:

Waterfall Model:


Analysis Phase
Design Phase
Coding Phase
Testing Phase

REQUIREMENTS ANALYSIS:

This first step is also the most important, because it involves gathering information about what the customer needs and defining, in the clearest possible terms, the problem that the product is expected to solve. Analysis includes understanding the customer's business context and constraints, the functions the product must perform, the performance levels it must adhere to, and the external systems it must be compatible with. Techniques used to obtain this understanding include customer interviews, use cases, and "shopping lists" of software features. The results of the analysis are typically captured in a formal requirements specification, which serves as input to the next step.

DESIGN:

This step consists of "defining the hardware and software architecture, components, modules, interfaces, and data...to satisfy specified requirements". It involves defining the hardware and software architecture, specifying performance and security parameters, designing data storage containers and constraints, choosing the IDE and programming language, and indicating strategies to deal with issues such as exception handling, resource management and interface connectivity. This is also the stage at which user interface design is addressed, including issues relating to navigation and accessibility. The output of this stage is one or more design specifications, which are used in the next stage of implementation.

IMPLEMENTATION:

This step consists of actually constructing the product as per the design specification(s) developed in the previous step. Typically, this step is performed by a development team consisting of programmers, interface designers and other specialists, using tools such as compilers, debuggers, interpreters and media editors. The output of this step is one or more product components, built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirements. For projects involving a large team, version control is recommended to track changes to the code tree and revert to previous snapshots in case of problems.

TESTING:

In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error-free and fully meet the requirements outlined in the first step. An independent quality assurance team defines "test cases" to evaluate whether the product fully or partially satisfies the requirements outlined in the first step. Three types of testing typically take place: unit testing of individual code modules; system testing of the integrated product; and acceptance testing, formally conducted by or on behalf of the customer. Defects, if found, are logged and feedback provided to the implementation team to enable correction. This is also the stage at which product documentation, such as a user manual, is prepared, reviewed and published.

INSTALLATION:

This step occurs once the product has been tested and certified as fit for use, and involves preparing the system or product for installation and use at the customer site. Delivery may take place via the Internet or physical media, and the deliverable is typically tagged with a formal revision number to facilitate updates at a later date.

MAINTENANCE:

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Typically, every change made to the product during the maintenance cycle is recorded and a new product release is performed to enable the customer to gain the benefit of the update.

3.2 ADVANTAGES:

The waterfall model, as described above, offers numerous advantages for software developers. First, the staged development cycle enforces discipline: every phase has a defined start and end point, and progress can be conclusively identified by both vendor and client. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage, or of customer expectations not being met.Getting the requirements and design out of the way first also

improves quality; it's much easier to catch and correct possible flaws at the design stage than at the testing stage, after all the components have been integrated and tracking down specific errors is more complex. Finally, because the first two phases end in the production of a formal specification, the waterfall model can aid efficient knowledge transfer when team members are dispersed in different locations.

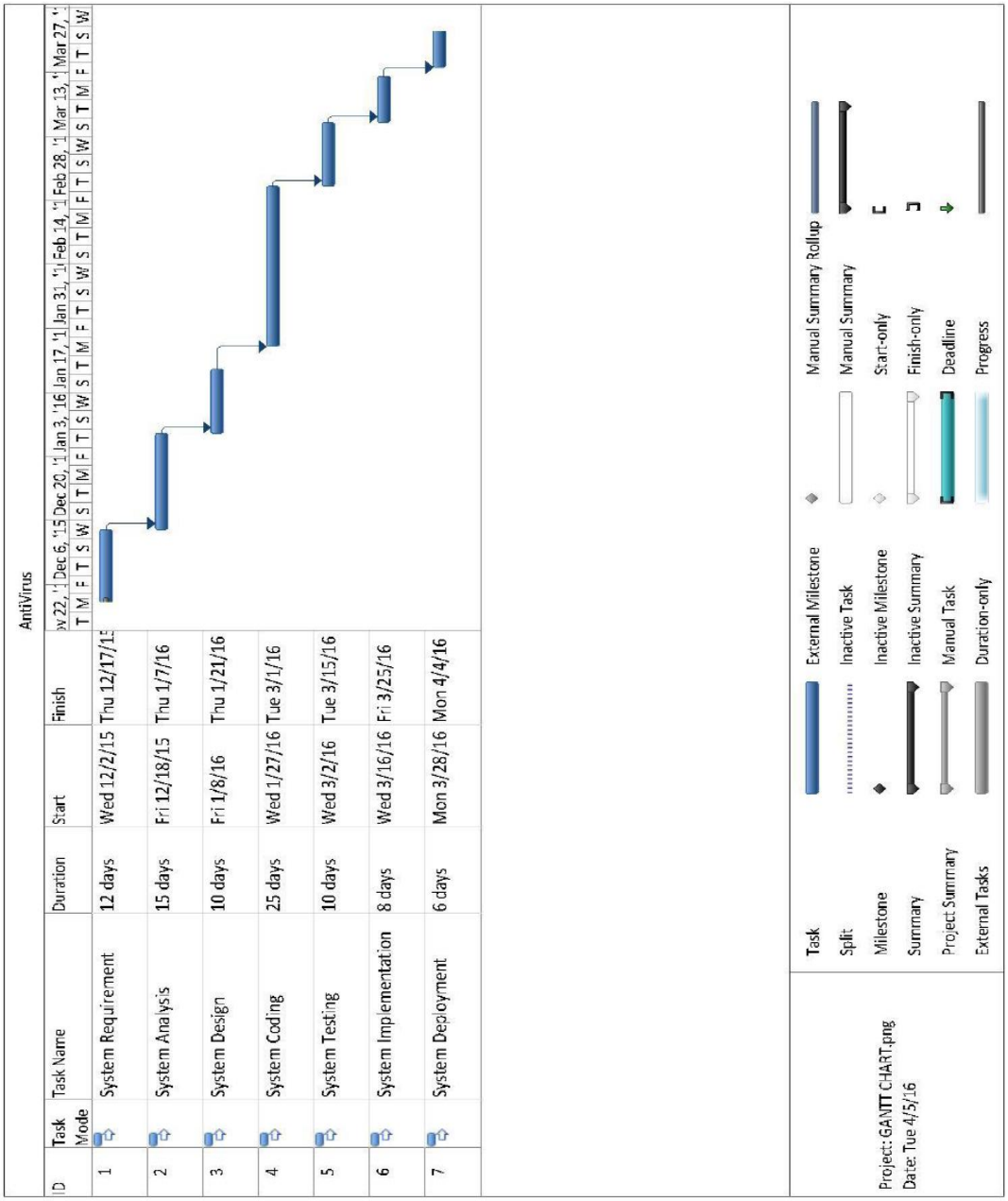
PROJECT SCHEDULE

GANTT CHART :

A Gantt chart is a horizontal bar chart used in project management as a tool for graphically representing the schedule of a set of specific activities or tasks. The horizontal bars indicate the length of time allocated to each activity, so the x-axis of a Gantt chart is subdivided into equal units of time, e.g. days, weeks, months. The y-axis of a Gantt chart. A simple look at a Gantt chart should enable its user to determine which tasks take the longest time to complete, which tasks are overlapping with each other, etc.

A Gantt chart indicates the following:

1) Duration and timelines of the listed activities.
2) The target and actual completion dates of the activities.
3) The cost of each activity;
4) The person or group of person responsible for each activity.
5) Milestones in the progress of the project.


AntiVirus
ID
Task Mode
Task Name
Duration
Start
Finish
1 System Requirement 12 days Wed 12/2/15 Thu 12/17/15
2 System Analysis 15 days Fri 12/18/15 Thu 1/7/16
3 System Design 10 days Fri 1/8/16 Thu 1/21/16
4 System Coding 25 days Wed 1/27/16 Tue 3/1/16
5 System Testing 10 days Wed 3/2/16 Tue 3/15/16
6 System Implementation 8 days Wed 3/16/16 Fri 3/25/16
7 System Deployment 6 days Mon 3/28/16 Mon 4/4/16
Project: GANTT CHART.png
Date: Tue 4/5/16
Task
Split
Milestone
Summary
Project Summary
External Tasks
External Milestone
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
Deadline
Progress

SYSTEM ANALYSIS

System Development can generally be thought of as having two major components: Systems Analysis and System Designs. Systems Design is the process

s of planning a new business system or one to replace or component an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective. System analysis the is the process of gathering and interpreting facts diagnosing problems, and using the information to recommend improvements to the system.

6.1 Feasibility Study:

It means to carry out the detailed study of the existing system, find out the problem related to technical, operational staff and economical field like cost by concerning the user of the study proposal of the proposed system is accepted by the management it will lead to the investigation of the existing system or problem area. Having recognized the problem , the next step to carry overall analysis of the system requirement in terms of its :

1. The input data
2. Type of processing needed
3. The output

Three Phases of the feasibility study is used:-

6.1.1 Technical Feasibility:

It is concerned with the available hardware and the software resources whether they meet the given requirement of the analysed system or not which include latest machinery and the technique required handling the system. It may also invoke the study of the new alternative to solve the given problem.

6.1.2Behavioral Feasibility:

People are inherently resistant to change, and computers have known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge installations have something to do with

turnover , transfers, restraining and changes in employee job status. Therefore it is understandable that the introduction of a candidate system requires special effort to educate , sell and train the staff on new ways of conducting business.

6.1.3Economical Feasibility:

It Deals with the study of the cost benefits analysis. All the cost of the new system compared with the benefits which can be obtained for management approval. The benefit may be quantities in nature Current System Summary.The genuine consideration of the system being developed is the approach we follow to look the system in the way it is useful for us.

SYSTEM DESIGN

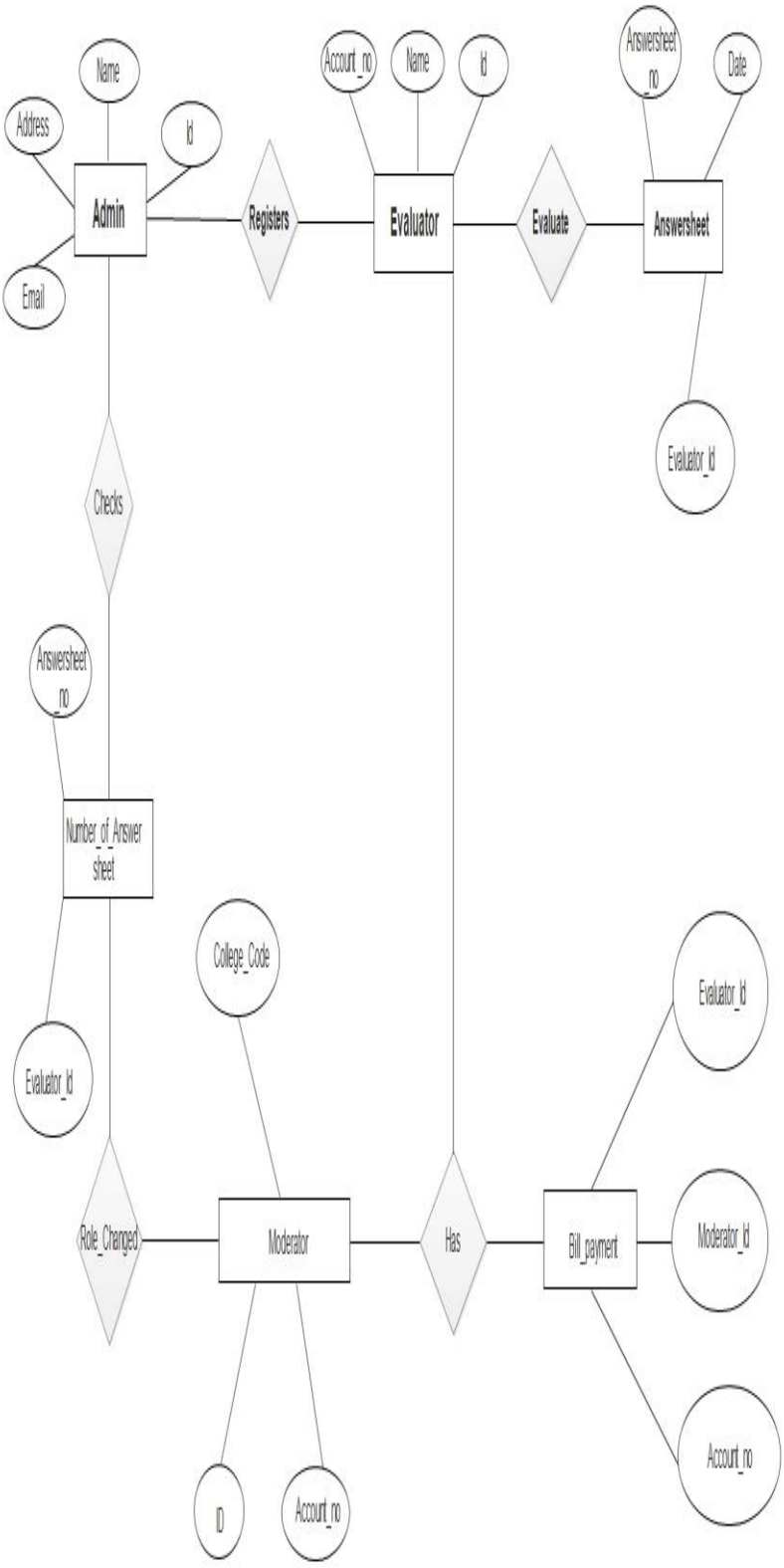
7.1 E-R Diagram

An entity-relationship (ER) diagram is a specialized graphic that represents the relationships between entities in database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to relationship and ovals are used represent attributes.

Types Of Database Relationship:

There three different types of database relationships, each named according to the number of tables rows that may be involved in the relationship. Each of these three relation types exists between two tables.

- **One-to-one relationships** occurs when each entry in the first table has one, and only one , counterpart in the second table. One-to-one relationships are rarely used because it is often more efficient to simply put all of the information in a single table.
- **One-to-many relationships** are the most common type of database relationship. They occur when each record in the first table corresponds to only one record in the first table.
- **Many-to-many relationship** occur when each record in the first table corresponds to one or more records in the second table and each record in the second table correspond to one or more records in the first table.

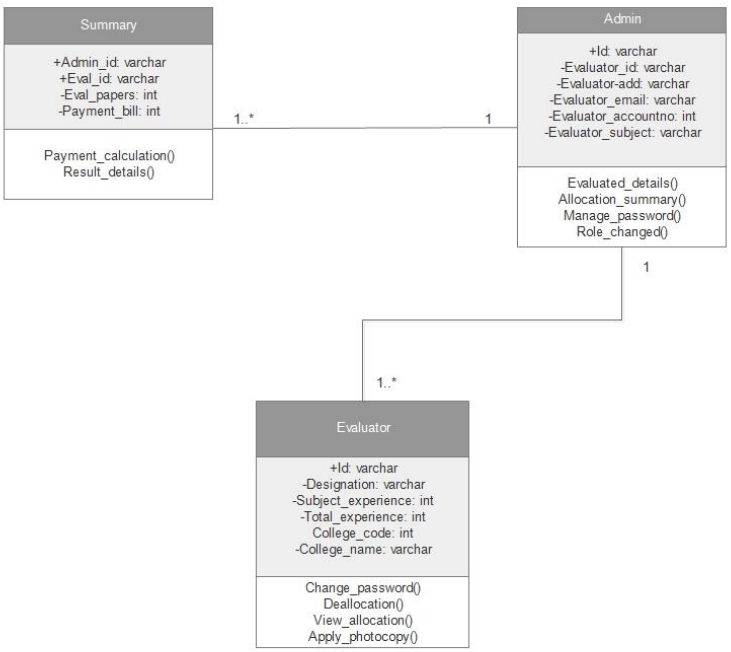

Name
Address
Id
Email
Admin
Registers
Account_no
Name
Id
Evaluator
Evaluate
Answersheet_no
Date
Answersheet
Evaluator_Id
Checks
Answersheet_no
Number_of_Answer sheet
Evaluator_Id
College_Code
Role_Changed
Moderator
Has
Bill_payment
Evaluator_Id
Moderator_Id
Account_no
ID
Account_no

7.2 Class Diagram

The class diagram is a static diagram. It represent the static view of an application. Class diagram is not only used for visualising, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describe the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

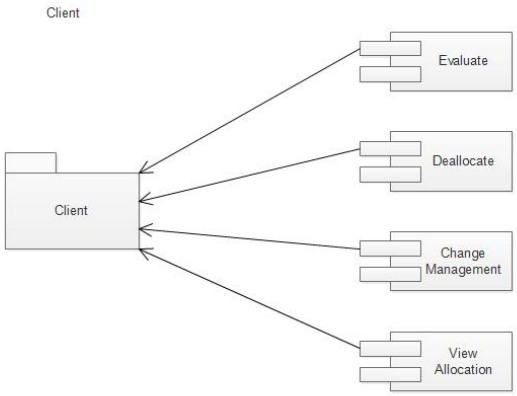
The class diagrams shows a collection of classes , interface, association, collaboration and constraints . It also known as a structural diagram.



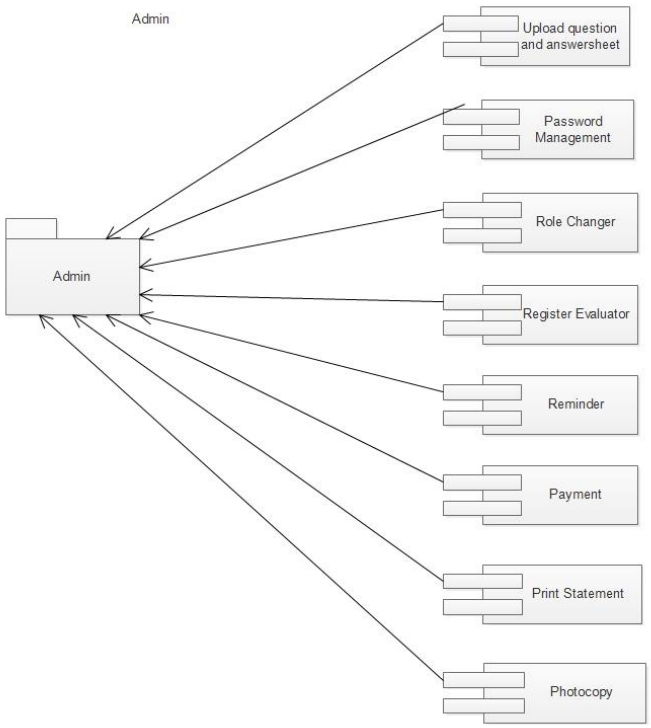
7.3 Component Diagram

A component defines its behaviour in terms of provided and required interfaces. Component are wired together by using as assembly connector to connect the required interfaces of one component with the provided interface of another component.
An assembly connector is a connector between two component that defines that one component provides the services that another component requires.
A component diagrams depicts how components are wired together to form larger components or software systems.
When using a component diagram to show the internal structure of a component , the provided and required interfaces of the encompassing component the provided and required interfaces of the contained components.

Client:



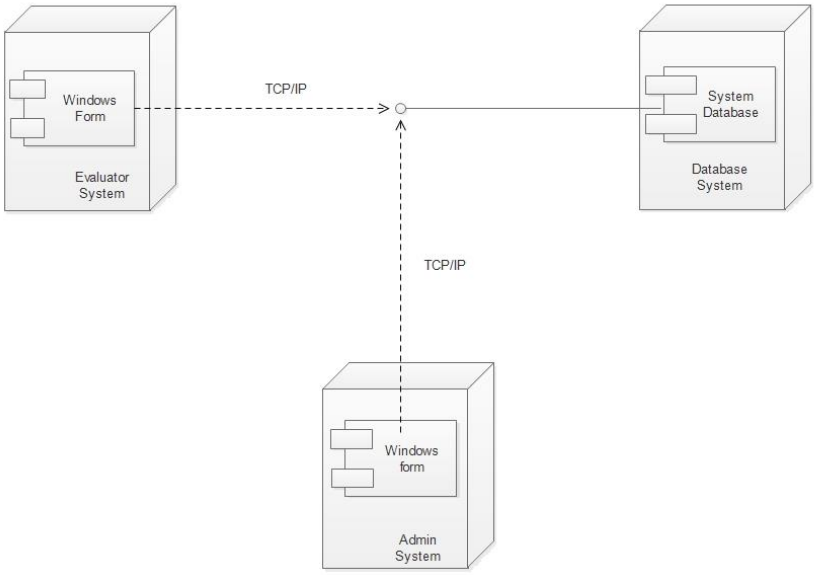
Server:


Admin
Admin
Upload question and answersheet
Password Management
Role Changer
Register Evaluator
Reminder
Payment
Print Statement
Photocopy

7.4 Deployment Diagram

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.
Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.
UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.
So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.
The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

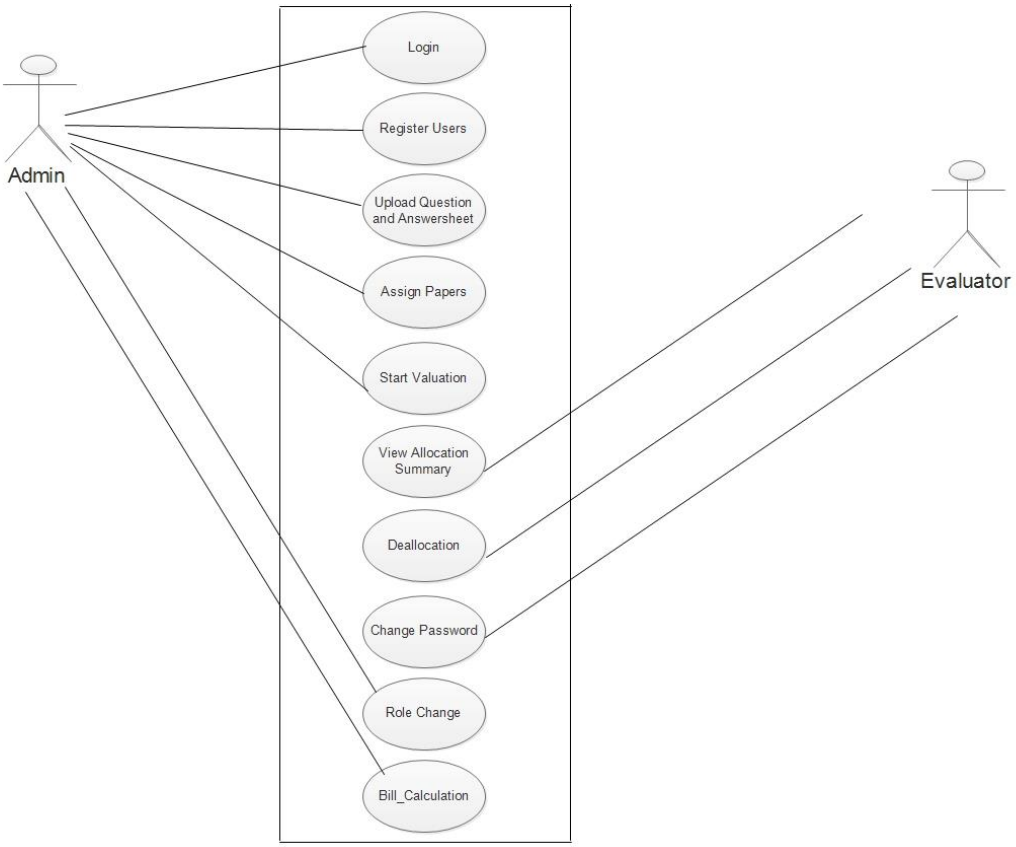


7.5 Use-Case Diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases.
So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.
The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw an use case diagram we should have the following items identified.

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

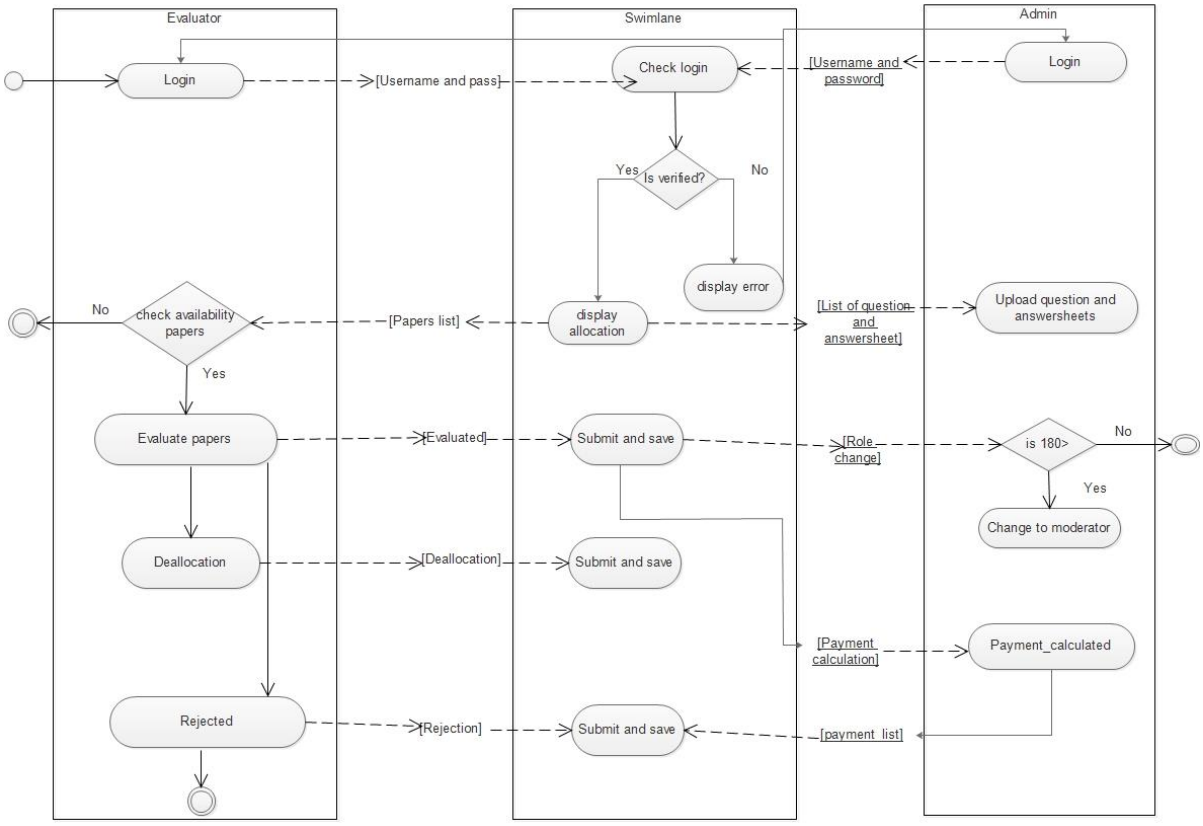
Use case diagrams are drawn to capture the functional requirements of a system.



7.6 Activity Diagram

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.
Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.
It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.
So the purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.



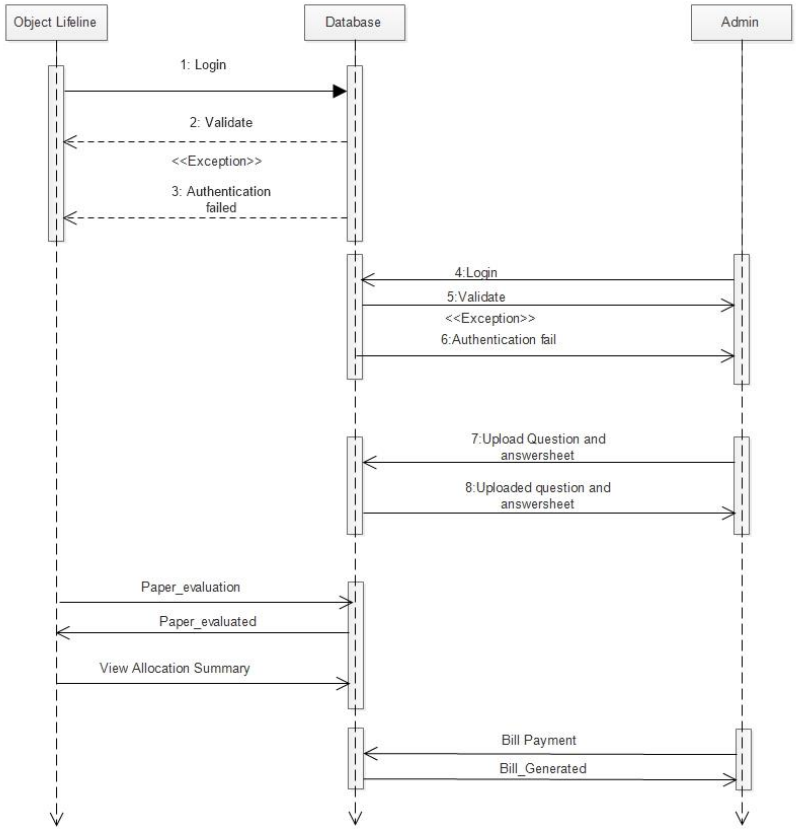
7.7 Sequence Diagram

The purposes of interaction diagrams are to visualize the interactive behaviour of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

So the purposes of interaction diagram can be describes as:

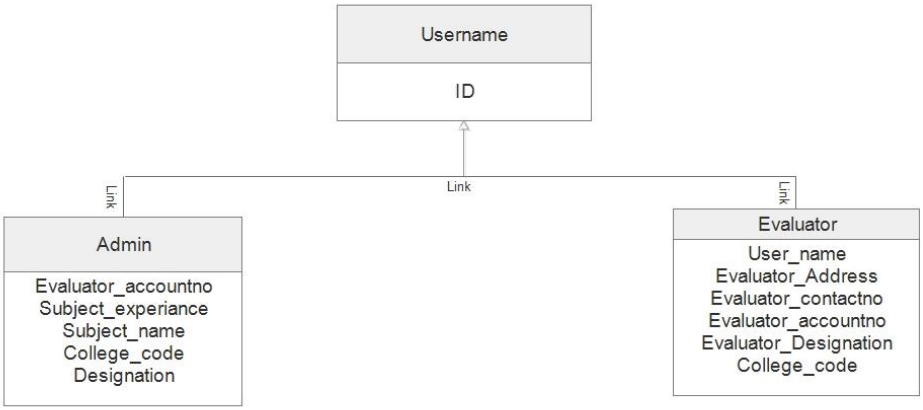
- To capture dynamic behaviour of a system
- To describe the message flow in the system
- To describe structural organization of the objects
- To describe interaction among objects.



7.8 Object Diagram

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams.
The difference is that a class diagram represents an abstract model consisting of classes and their relationships. But an object diagram represents an instance at a particular moment which is concrete in nature.
It means the object diagram is more close to the actual system behaviour. The purpose is to capture the static view of a system at a particular moment.
So the purpose of the object diagram can be summarized as:

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behaviour and their relationship from practical perspective

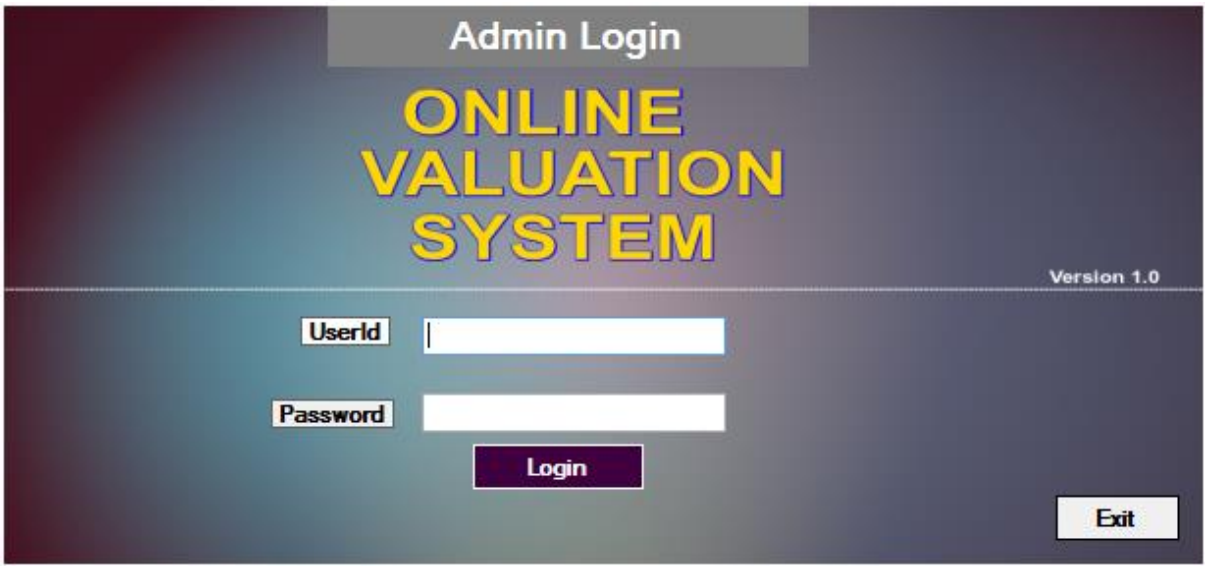


IMPLEMENTATION

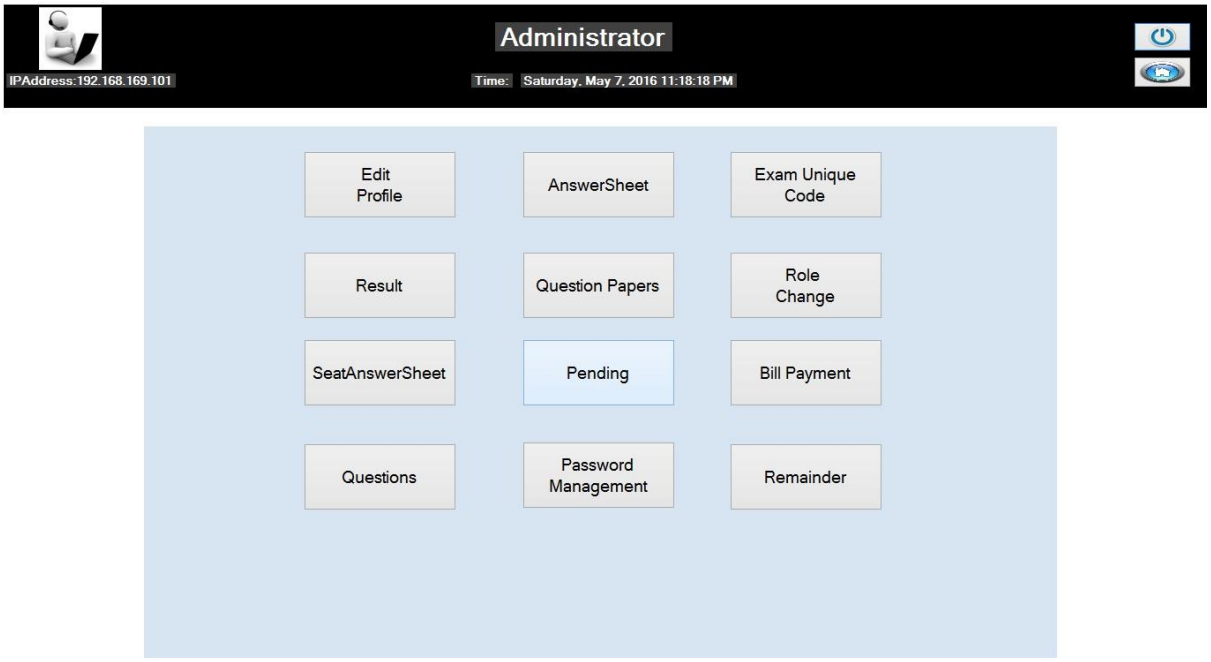
GRAPHICAL USER INTERFACE :

Server Side:

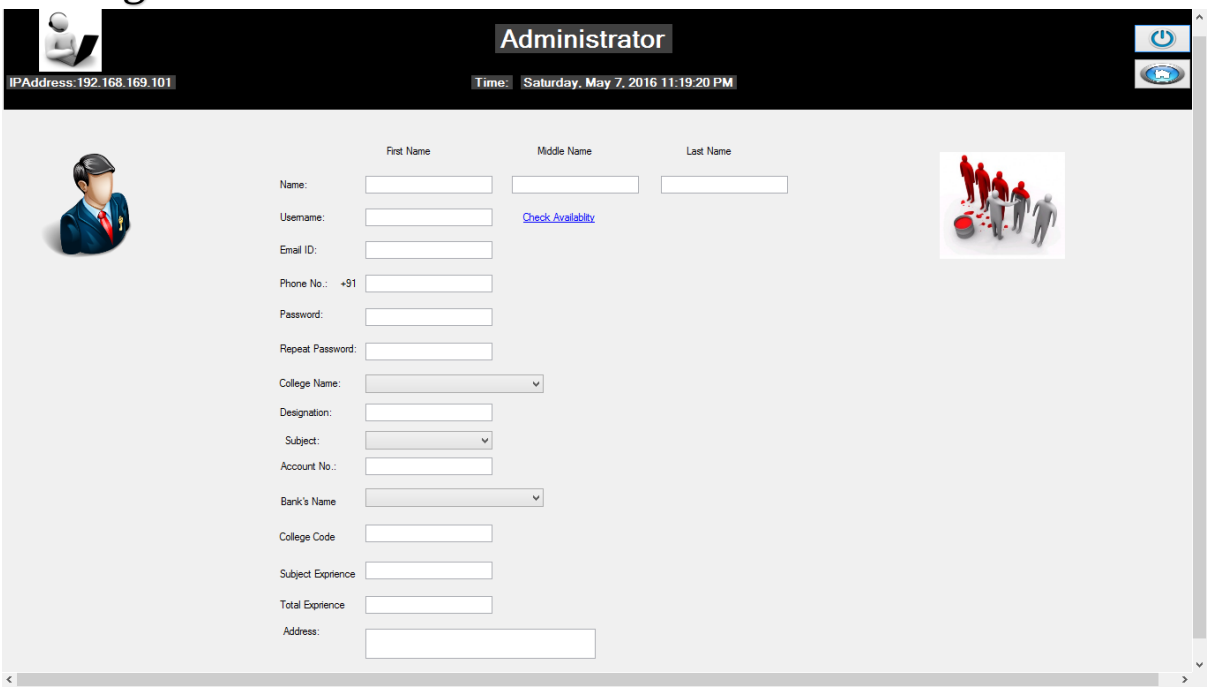
Admin Login Page



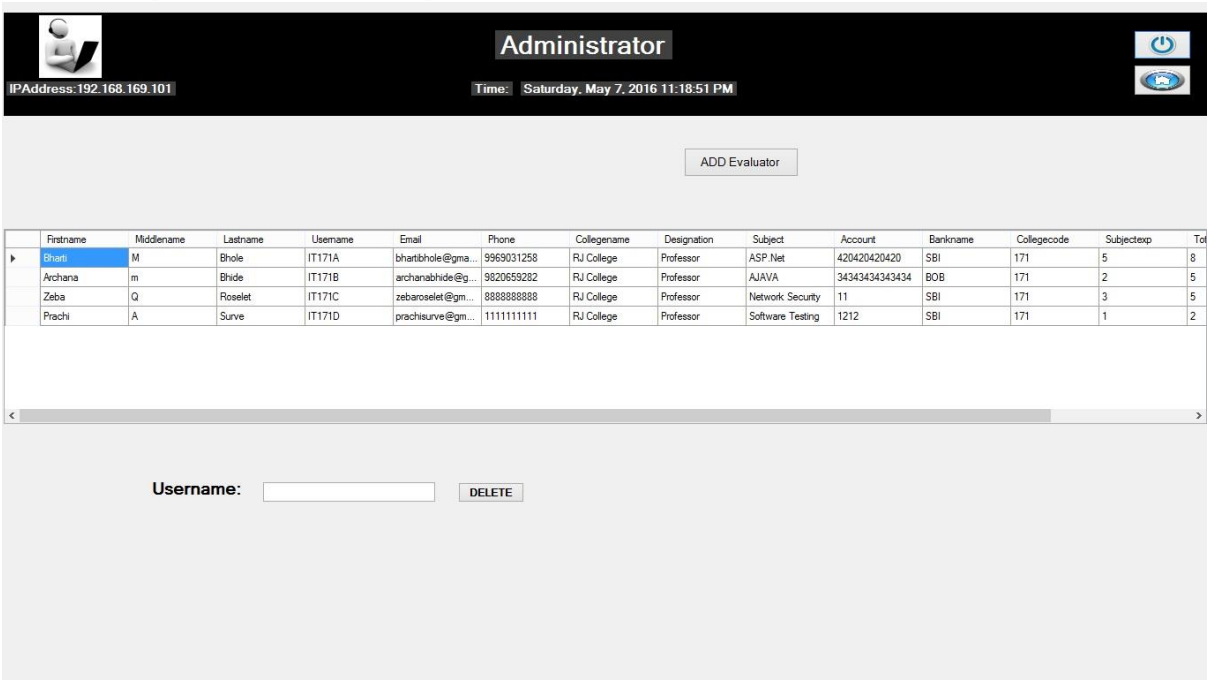
Main Page


Administrator
IPAddress:192.168.169.101
Time: Saturday, May 7, 2016 11:18:18 PM
Edit Profile
AnswerSheet
Exam Unique Code
Result
Question Papers
Role Change
SeatAnswerSheet
Pending
Bill Payment
Questions
Password Management
Remainder

Adding Evaluators

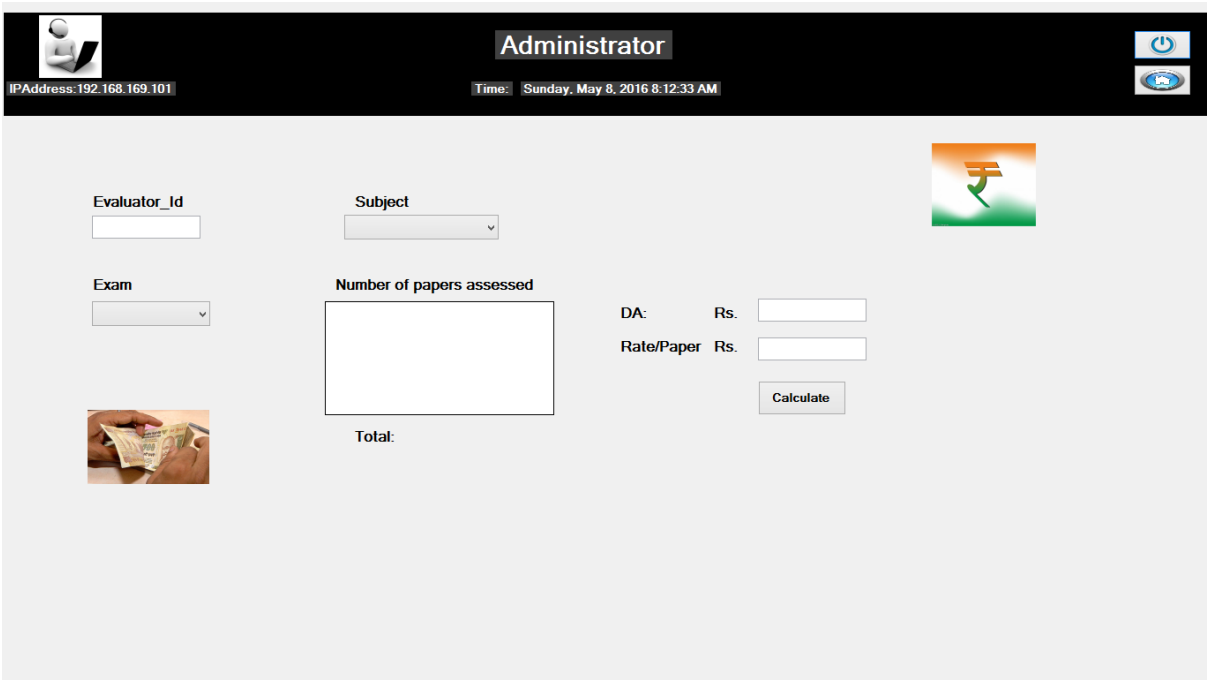


Viewing And Deleting Evaluators

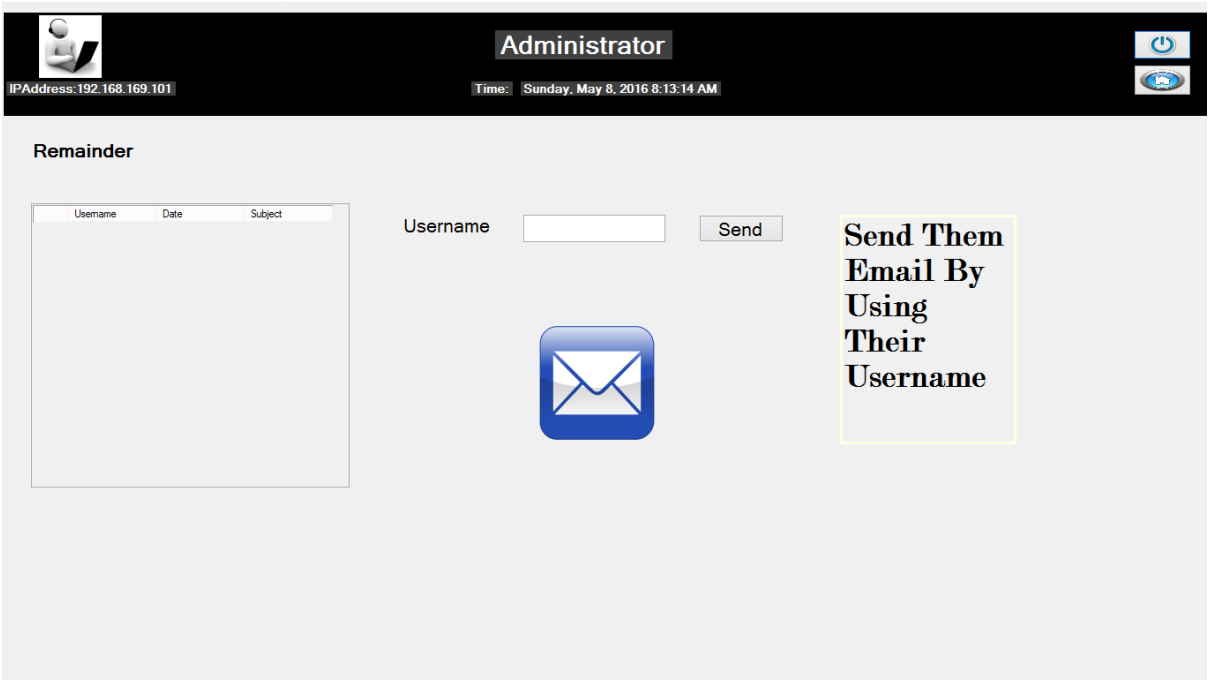


Firstname	Middlename	Lastname	Username	Email	Phone	Collegename	Designation	Subject	Account	Bankname	Collegecode	Subjectexp	Tot
Bharti	M	Bhole	IT171A	bhartibhole@gma...	9969031258	RJ College	Professor	ASP.Net	420420420420	SBI	171	5	8
Archana	m	Bhide	IT171B	archanabhide@g...	9820659282	RJ College	Professor	AJAVA	34343434343434	BOB	171	2	5
Zeba	Q	Roselet	IT171C	zebaroselet@gm...	8888888888	RJ College	Professor	Network Security	11	SBI	171	3	5
Prachi	A	Surve	IT171D	prachisurve@gm...	1111111111	RJ College	Professor	Software Testing	1212	SBI	171	1	2

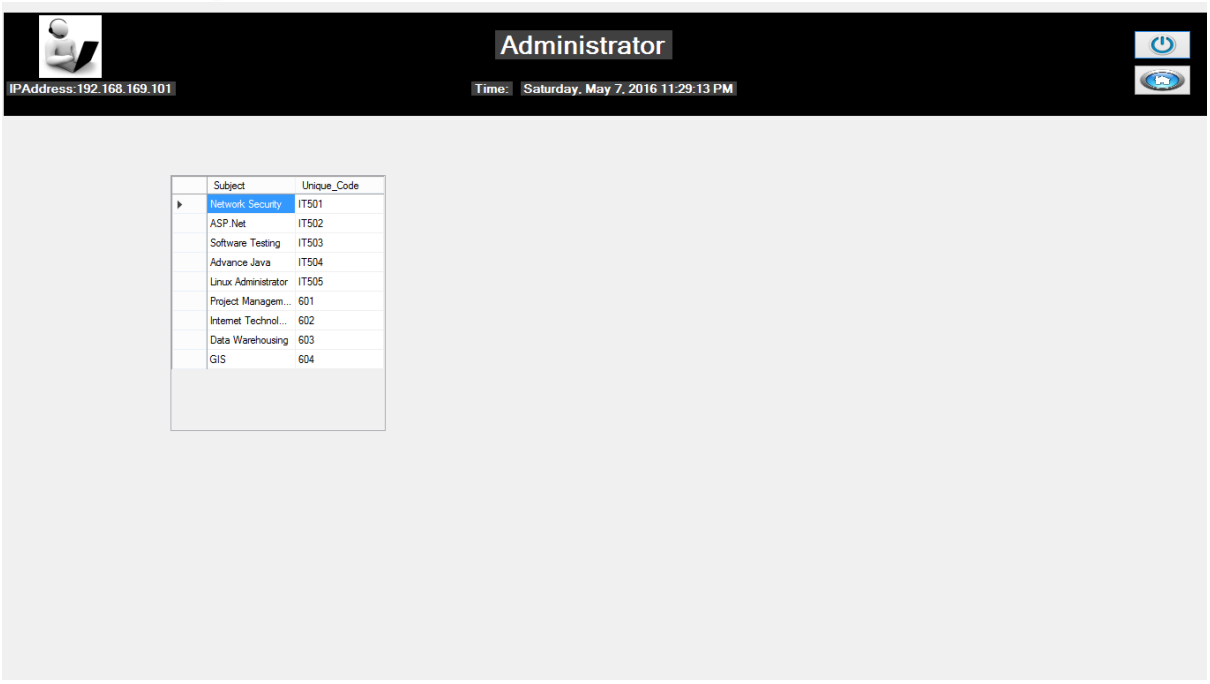
Bill Calculation



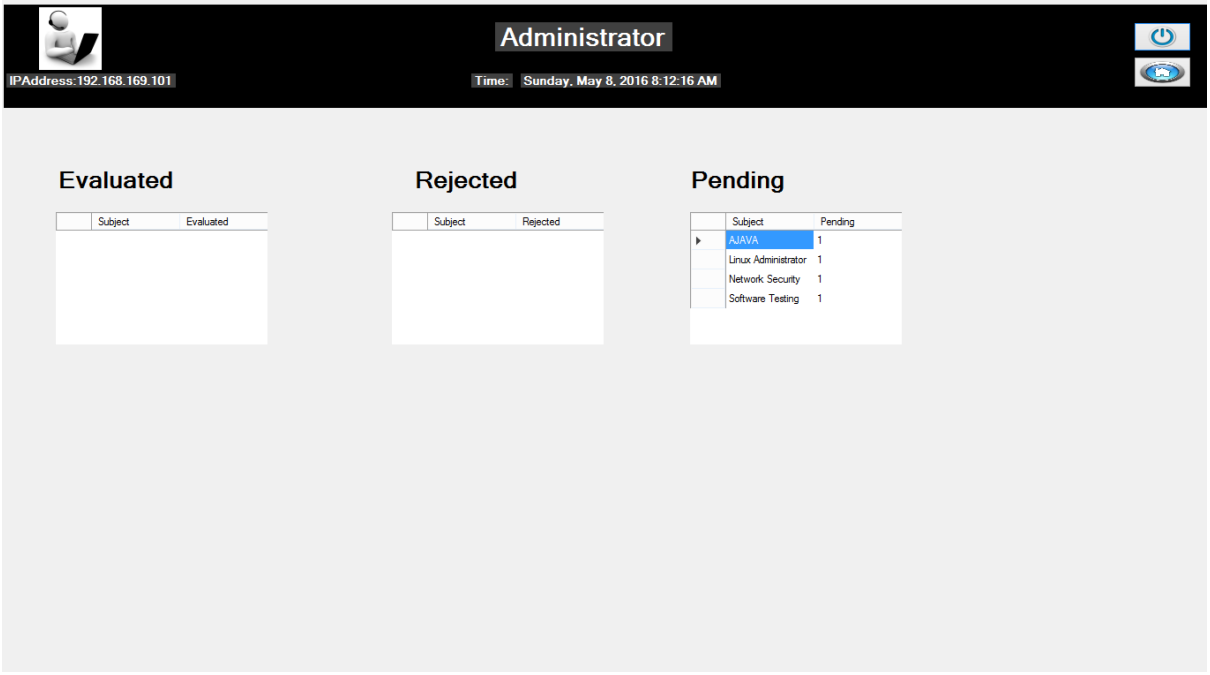
Remainder Page



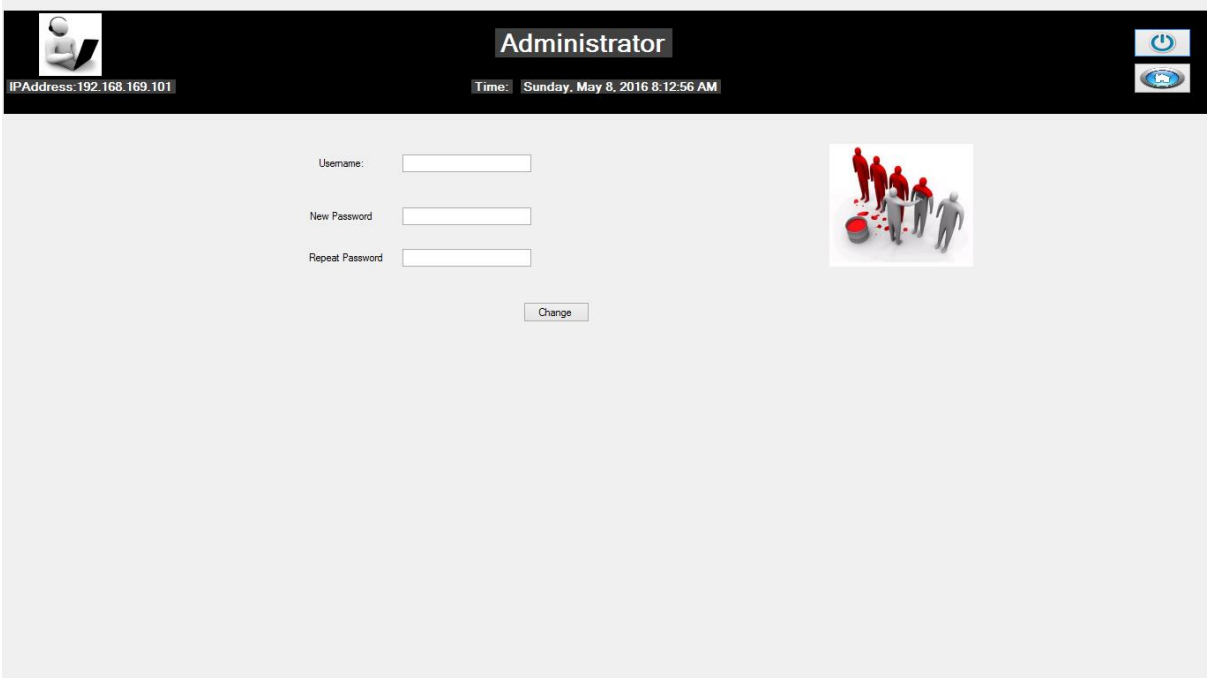
Exam Unique Code



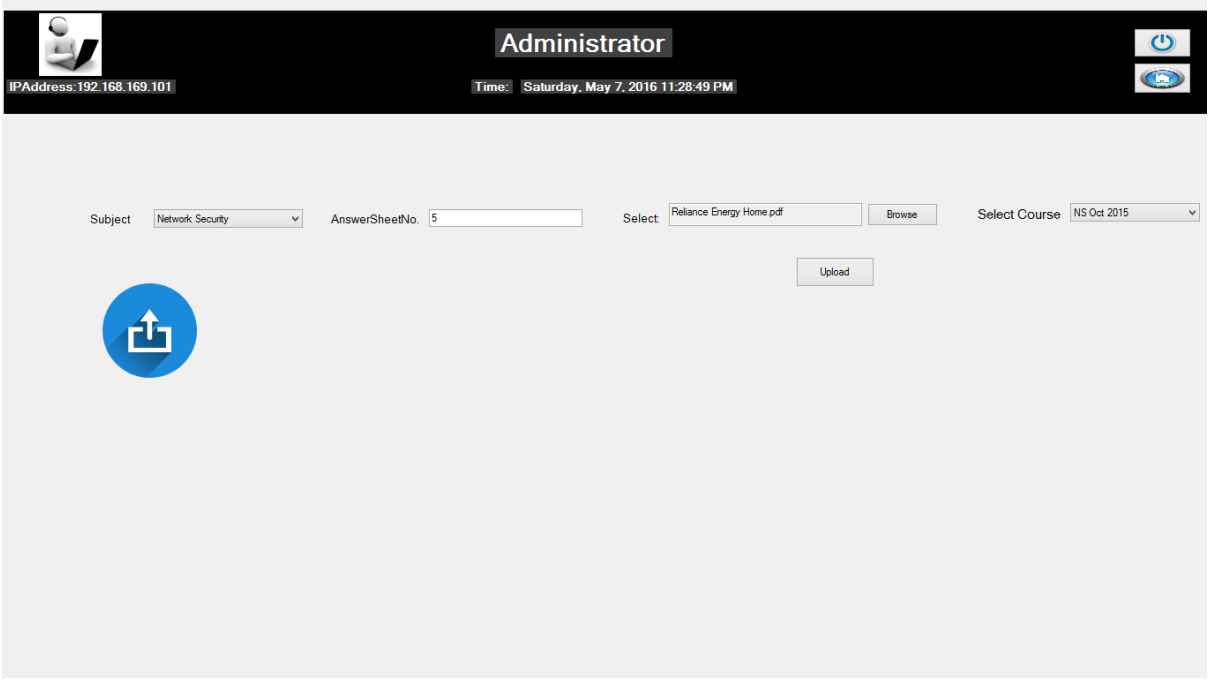
Status Page


Administrator
IPAddress:192.168.169.101
Time: Sunday, May 8, 2016 8:12:16 AM
Evaluated
Subject
Evaluated
Rejected
Subject
Rejected
Pending
Subject
Pending
AJAVA
1
Linux Administrator
1
Network Security
1
Software Testing
1

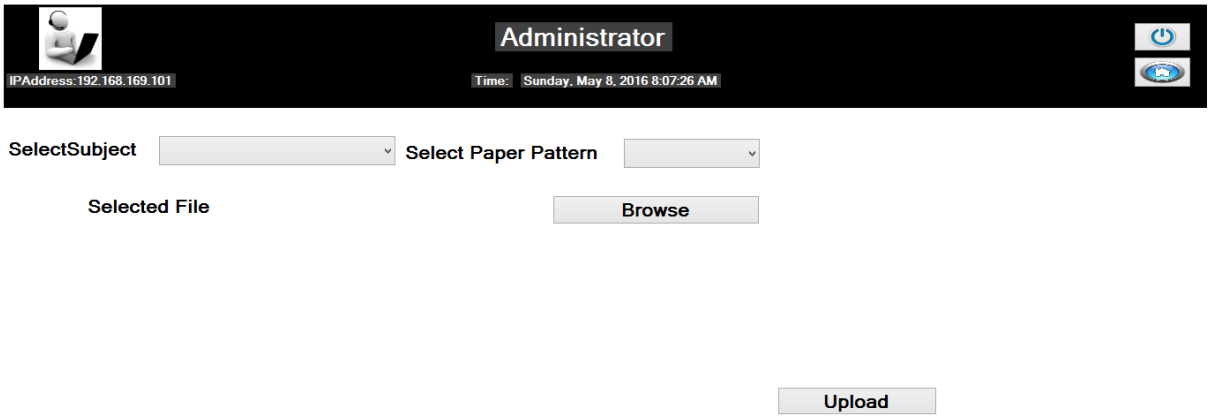
Password Changing



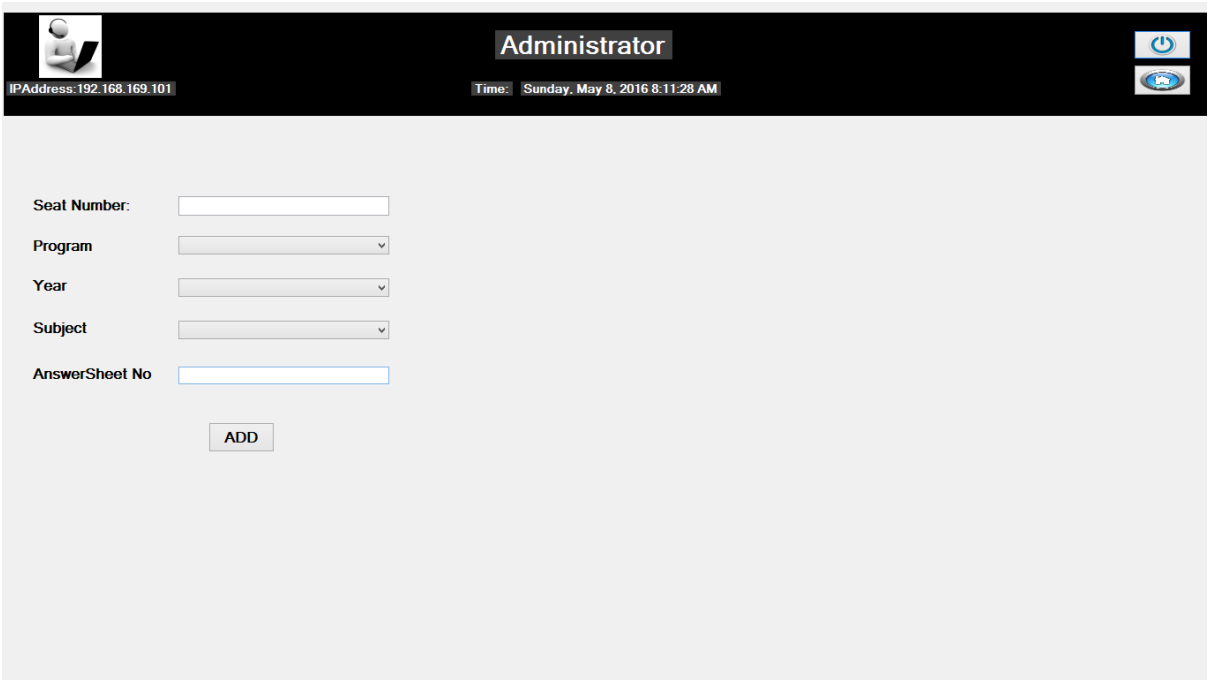
AnswerSheet Uploading



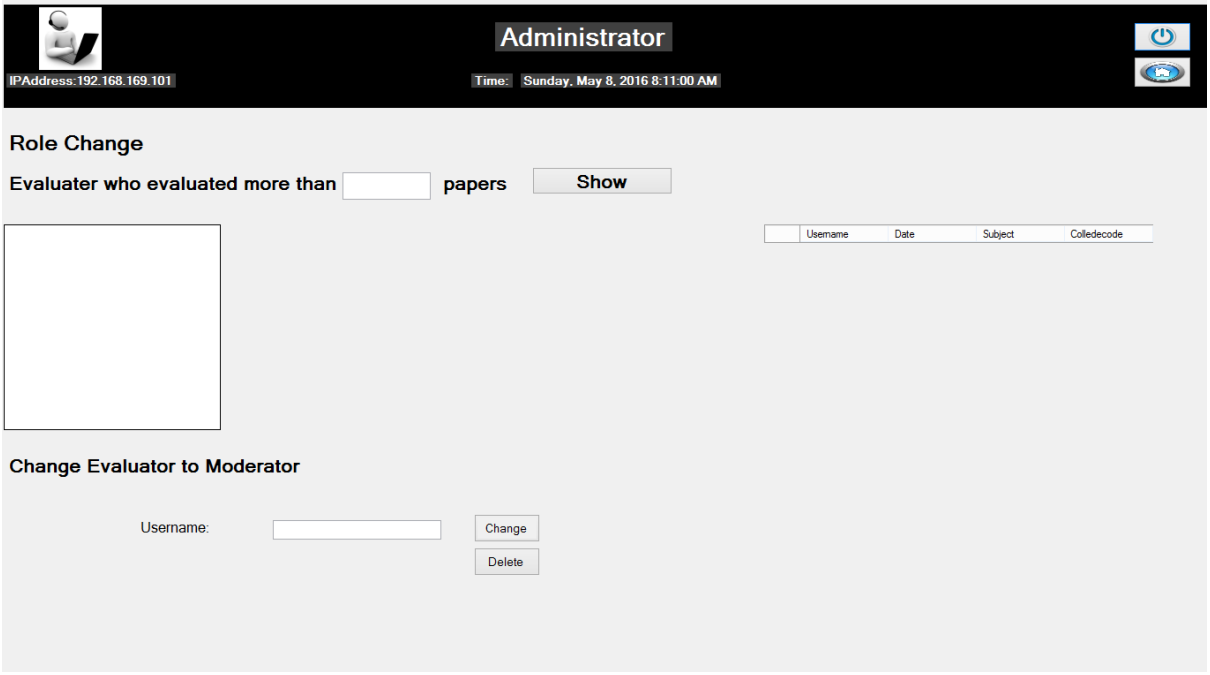
Question Paper Uploading



Student Seat Number And Answersheet Mapping

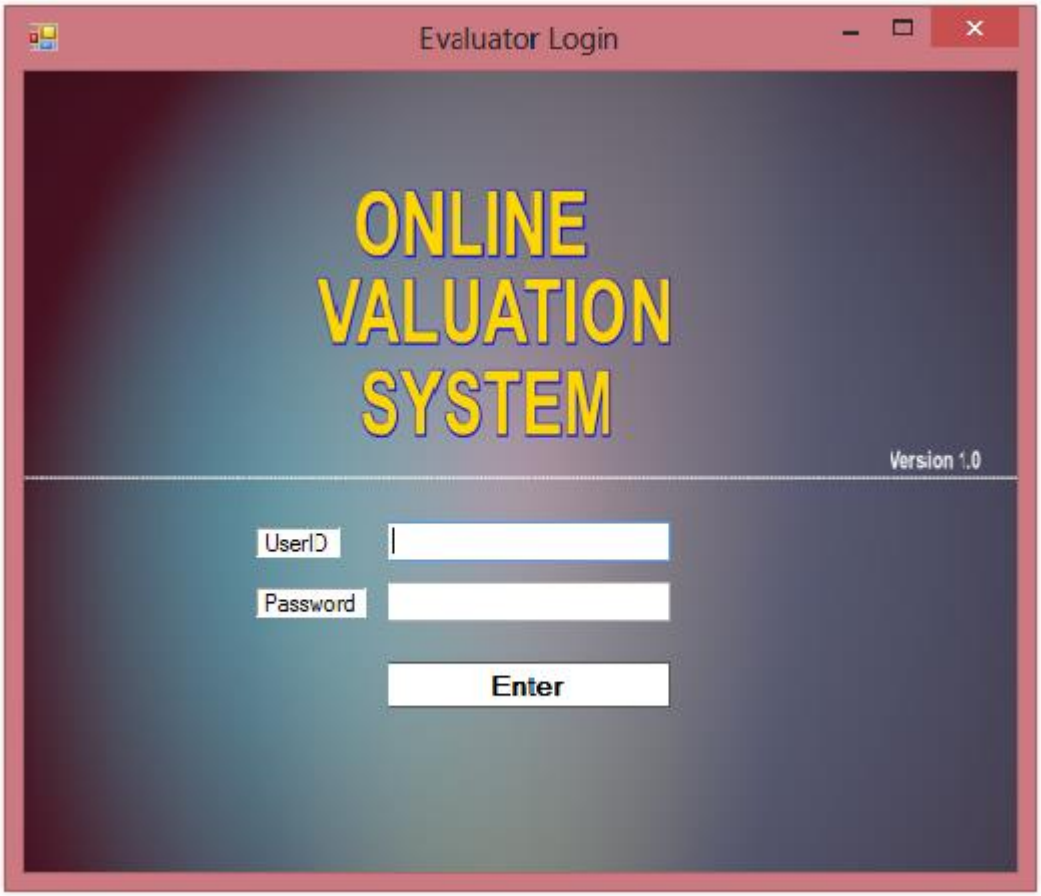


Role Changing

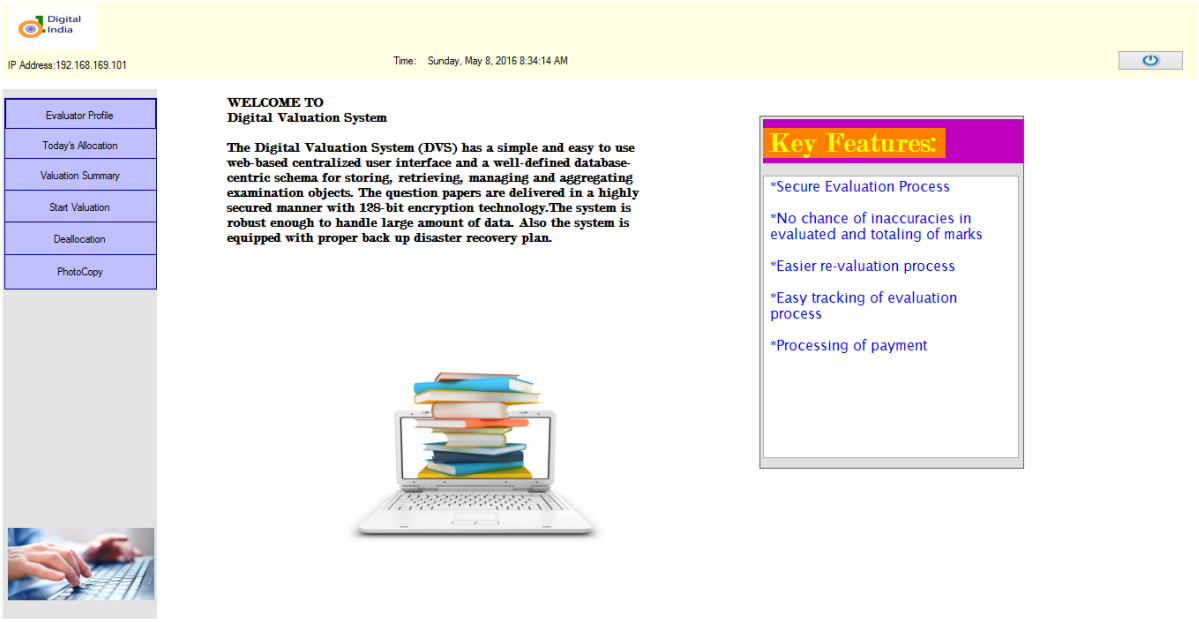


Evaluator Side:

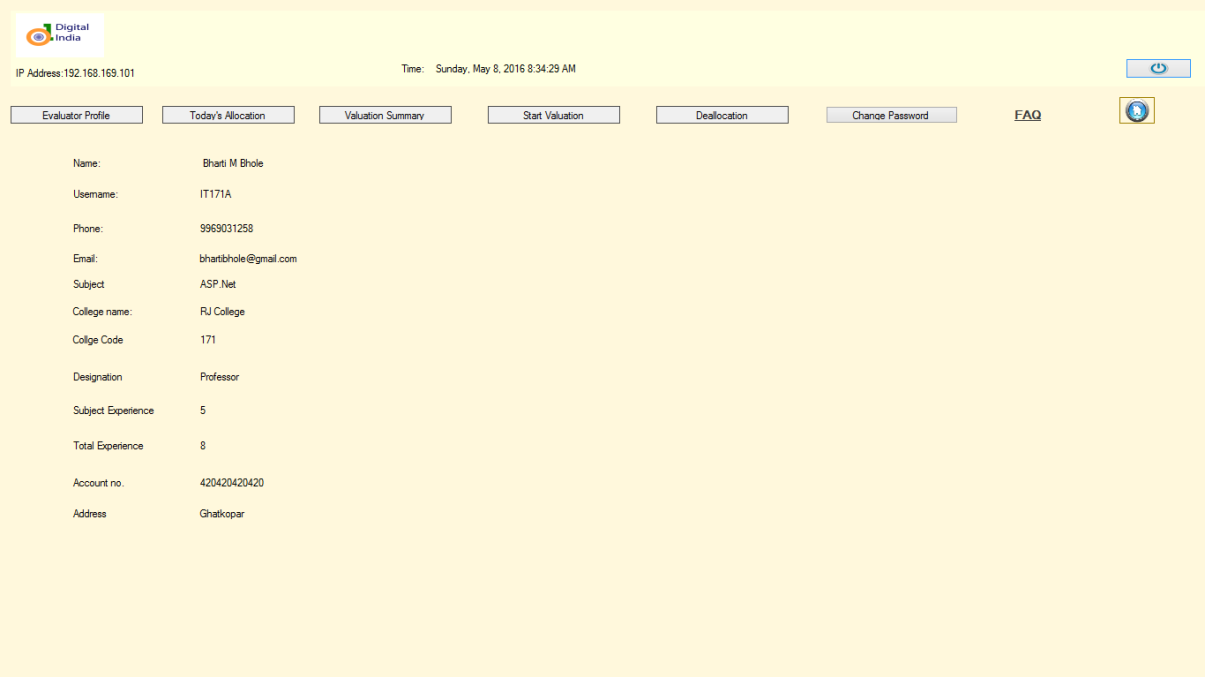
Login Screen(Evaluator)



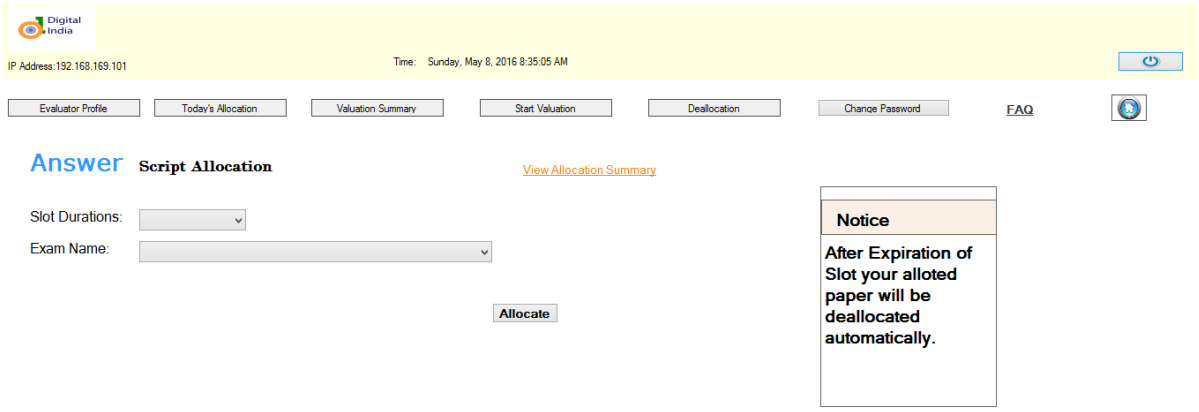
Main Screen


Digital India
IP Address:192.168.169.101
Time: Sunday, May 8, 2016 8:34:14 AM
Evaluator Profile
Today's Allocation
Valuation Summary
Start Valuation
Deallocation
PhotoCopy
WELCOME TO
Digital Valuation System
The Digital Valuation System (DVS) has a simple and easy to use web-based centralized user interface and a well-defined database-centric schema for storing, retrieving, managing and aggregating examination objects. The question papers are delivered in a highly secured manner with 128-bit encryption technology.The system is robust enough to handle large amount of data. Also the system is equipped with proper back up disaster recovery plan.
Key Features:
*Secure Evaluation Process
*No chance of inaccuracies in evaluated and totaling of marks
*Easier re-valuation process
*Easy tracking of evaluation process
*Processing of payment

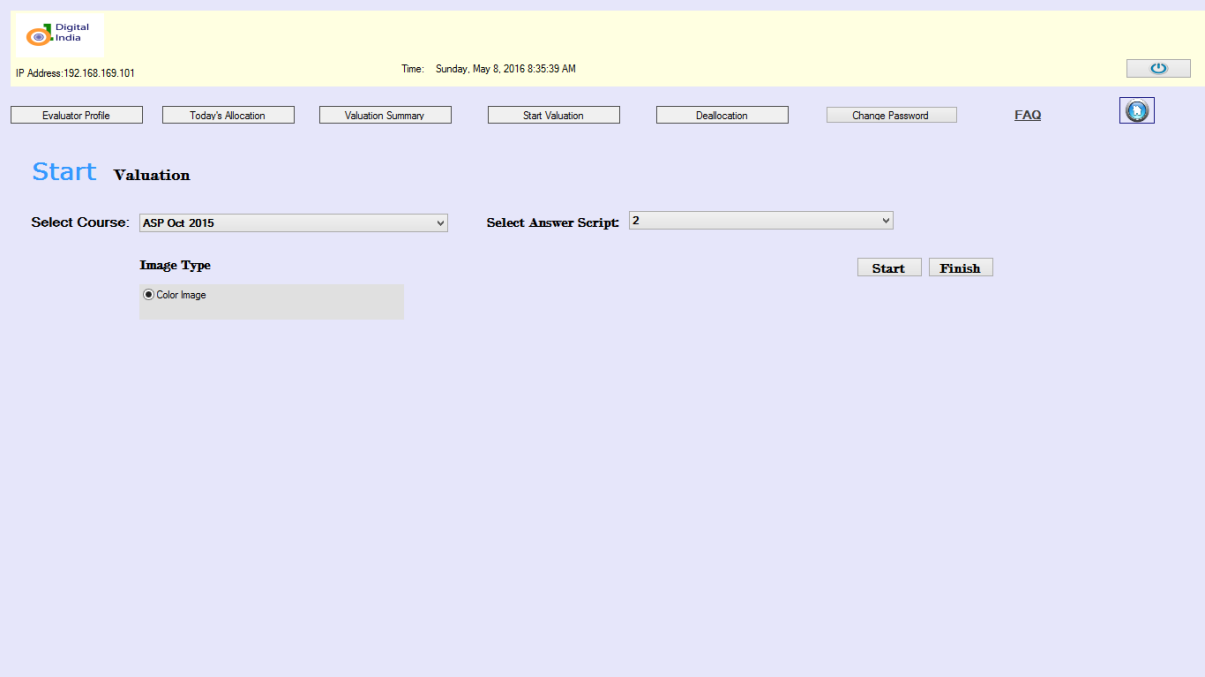
Evaluator Profile



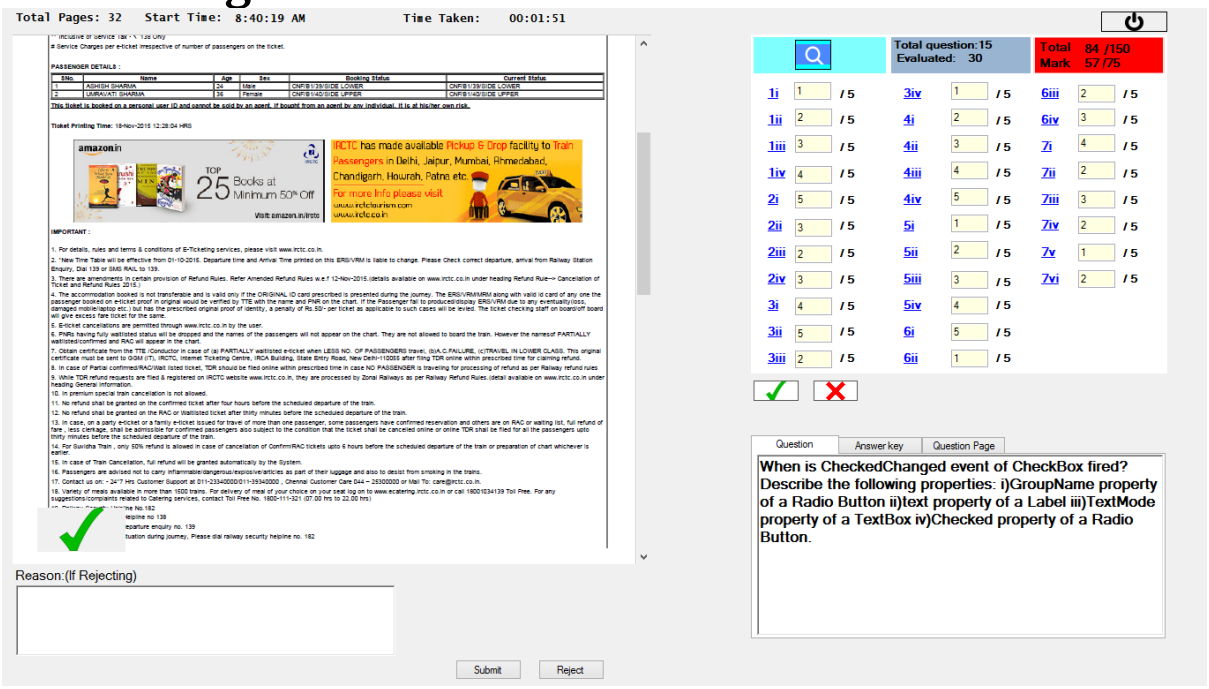
Todays Allocation



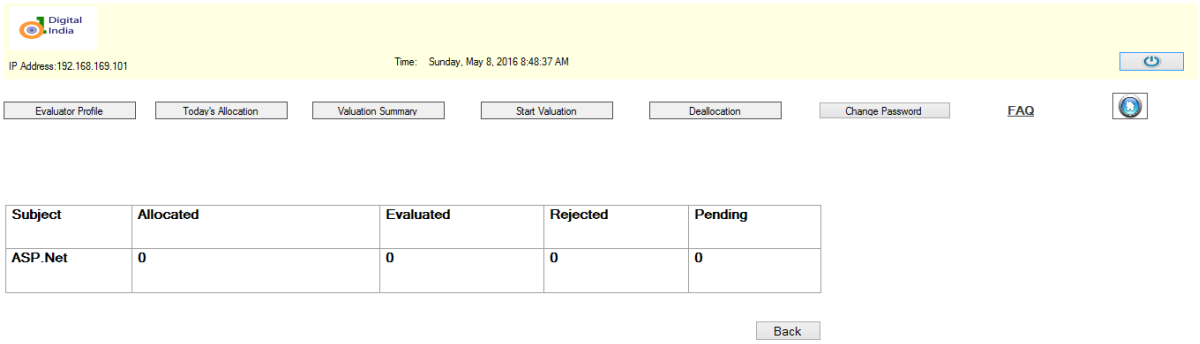
Start Valuation



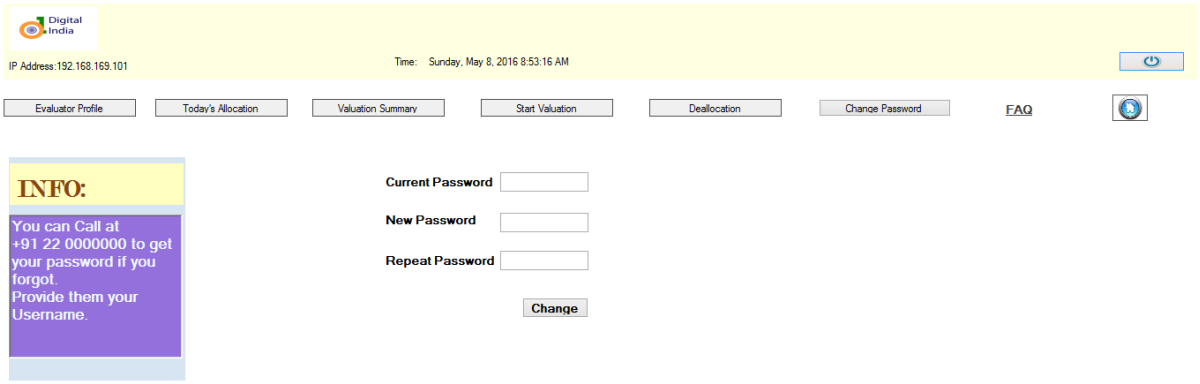
Evaluating Screen



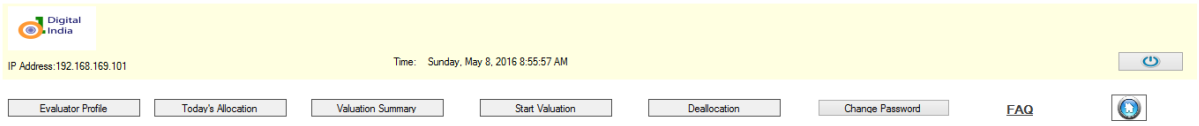
Deallocation



Password Changing



Allocation Summary



Exam BSC(IT)

Subject ASP.Net

Date 5/8/2016

Date	Evaluated	Rejected	Deallocated
5/8/2016 1	1	0	0

DEPLOYMENT

Software deployment is all of the activities that make a software system available for use. The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the producer site or at the consumer site or both. Because every software system is unique, the precise processes or procedures within each activity can hardly be defined. Therefore, "deployment" should be interpreted as a general process that has to be customized according to specific requirements or characteristics.
We can deploy ASP.NET Application in 3 different ways:

- xCopy Deplyment
- PreCompiled Deployemnt
- Web Setup Project

The choice of best deployment alternative depends upon particular need of each application. xCopy deployment is easiest , and it is often used during development to create copies of application on different servers for testing purpose. For small application xCopy deployment may be the best choice.
Precompiled deployment has several advantages over XCopy deployment. E.g. precompiled deployment is always gives better performance for the first users of the site at the same time it is more secure as we don't need to copy our source code files on to server.
If our application deployed on one or few servers then precompiled deployment is usually best choice.
When we are going on deploy our application on number of servers then creating a setup program is a very handy tool. Although creating this setup program is much tedious and involves considerable working, the deployment form this setup program becomes very easier.
In our project we have Precompiled Deployment for deploying Online Job Portal System application.

TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Strategic Approach To Software Testing:

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Taking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

White Box Testing:

This type of testing ensures that

- All loops are executed at their boundaries and within their operational
- All independent path have been excerside atleast once.

- All internal structure have been exercised to assure their validity

To follow the concept of white box testing we have tested each form .We have created independently to verify that Data flow is correct, all conditions are exercised to check their validity, all loops are executed on their boundaries.

Basic Path Testing:

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions.

The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V (G) =E-N+2

Or

V (G) =P+1

Or

V (G) = Number Of Regions

Where V (G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent path

Conditional Testing:

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

Data Flow Testing:

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

Loop Testing:

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

All the loops were tested at their limits, just above them and just below

All the loops were skipped at least once.

For concatenated loops the values of dependent loops were set with the help of connected loop.

MAINTENANCE & FUTHER ENHANCEMENT

Since the software is robust and will work properly and efficiently. There are certain steps or conditions that the user must look into, so the lifetime of the software can be increased and thus already maintained. When system is allowed to run for sometimes they tend to become disorganized.
This results in system inefficiently. To provide a course of action, where by the inefficiency in the system is arrested and state of the system is brought back to equilibrium, we undertake certain measure. This is done with the help of maintenance of the system.
The maintenance of the system is very much essential in the long run of the system. This maintenance can only be carried out if a proper evaluation of the system is carried our if a proper evaluation of the system is carried out if a proper evaluation of the system is carried out if a proper maintenance of the system is carried out at regular intervals. So the schemes for maintenance of the system are to be pre-determined before the actual implementation of the system.
Some of those conditions are as follows:

Always work with support of higher authority
Always keep other windows close while on this software
Read the user manual carefully before using the software
A periodic review of system at regular interval
A meeting with the client to access the current utility of the system and level of satisfaction
Bringing out system modification.
Problems in the system arise due to business environment changes, leading to change in demand and wants of the client. Such a change in the requirement factors call for modification of the system to revised information.
Broadly we can state the major problems can be as follows:
Making the required modification to suit
As the business environment has changed the system needs to be changed as well.

The four types of maintenance activities are describe below:

CORRECTIVE MAINTENANCE

Corrective maintenance is the most commonly used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometime it causes spreading of damage to other parts. In most cases this is costly business. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of system maintained by this type of maintenance is not known and cannot be measured. Therefore, corrective maintenance is carried out on all items where the consequences of failure or wearing out are not significant and the cost of this maintenance is not greater than preventive maintenance.

ADAPTIVE MAINTENANCE

Adaptive maintenance is an activity that modifies software to properly interface with the changing environment.

PERFECTIVE MAINTENANCE:

Perfective maintenance is performed to satisfy user requests such as new Capabilities, modification to existing function and general enhancements.

PREVENTIVE MAINTENANCE:

Preventive maintenance occurs when the software is changed to improve future maintainability or to provide a better basic for future enhancements.

FURTHER ENHANCEMENT:

There are many things to be enhance in this system:-

1) Questions are added manually so it can be scan and added automatically.
2) Result is just displayed,in further version it will have print option.
3) Photocopy option will be added on both side.
4)More security will be enforced on system.
5)Purformance will be enhanced.

CONCLUSION

The Software “DIGITAL VALUATION SYSTEM” has been developed in windows8 Professional Environment using C# as front end and back end.
It was a wonderful and learning experience for us while working on this project. This Project took us through the various phases of project development and gave us real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave us a feel of developers industry.
It was due to this project we came to know how professional softwares are designed.
We enjoyed each and every bit of work we had put into this project. The project is further extendable.

BIBLIOGRAPHY

Books:

- Pro C# and .NET 4.5 Framework
- Microsoft Visual Basic 2015 Step by Step by Michael Halvorson

WebSites:

- http://www.codeproject.com/
- http://stackoverflow.com/
- https://msdn.microsoft.com/en-us/library/